Homework 3: Mining Data Streams

Submit Assignment

Due Monday by 11:59pm **Points** 5 **Submitting** a file upload

The homework can be done in a group of 2 students.

Submission on time, i.e. before the deadline, is awarded with 3 exam bonus points if your homework is accepted, i.e. you have successfully presented and demonstrated your homework to a course teaching assistant during a presentation session.

Introduction

The following three papers present streaming graph processing methods and corresponding algorithms that make use of the stream mining algorithms presented in the course (Lectures 5-6 and **Chapter 4 (http://infolab.stanford.edu/~ullman/mmds/ch4.pdf)** of the text book *Mining of Massive Datasets*), namely, the reservoir sampling (Lecture 5) and the Flajolet-Martin algorithm to estimate the number of distinct elements in a stream (Lecture 6). The graph algorithms presented in first two papers make use of the reservoir sampling technique; whereas the graph algorithm in the third paper makes use of the Flajolet-Martin algorithm (HyperLogLog counters).

1. M. Jha, C. Seshadhri, and A. Pinar, **A Space-Efficient Streaming Algorithm for Estimating Transitivity and Triangle Counts Using the Birthday Paradox (https://arxiv.org/pdf/1212.2264.pdf)**, ACM TKDD, 9-3, 2015.
2. L. De Stefani, A. Epasto, M. Riondato, and E. Upfal, **TRIÈST: Counting Local and Global Triangles in Fully-Dynamic Streams with Fixed Memory Size (http://www.kdd.org/kdd2016/papers/files/rfp0465-de-stefaniA.pdf)**, KDD'16.
3. P. Boldi and S. Vigna, **In-Core Computation of Geometric Centralities with HyperBall: A Hundred Billion Nodes and Beyond (https://arxiv.org/pdf/1308.2144v2.pdf)**, ICDMW'13.

Task

You are to study and implement a streaming graph processing algorithm described in one of the above papers of your choice. In order to accomplished your task, you are to perform the following two steps

1. First, implement the reservoir sampling or the Flajolet-Martin algorithm used in the graph algorithm presented in the paper you have selected;

2. Second, implement the streaming graph algorithm presented in the paper that make use of the algorithm implemented in the first step.

To ensure that your implementation is correct, you are to test your implementation with some of the publicly available graph datasets (find a link below), and present your test results in a report.

Implementation can be done using any data processing framework that includes support for stream (streaming graph) processing such as Apache Spark, Apache Flink, or no framework, e.g., in Java, Python or other language of your choice.

Optional task for extra bonus

In your report, answer the following questions:

1. What were the challenges you have faced when implementing the algorithm?
2. Can the algorithm be easily parallelized? If yes, how? If not, why? Explain.
3. Does the algorithm work for unbounded graph streams? Explain.
4. Does the algorithm support edge deletions? If not, what modification would it need? Explain.

Datasets

Graph datasets can be found by following **this link** **(https://snap.stanford.edu/data/)**. (see in particular, datasets under "**Web graphs** **(https://snap.stanford.edu/data/#web)**") Note that information about each dataset (you can see it when you click on a dataset name) includes several metrics, e.g., Triangle Count, that you can use to verify your results.

Readings

- **Lecture 5: Mining Data Streams 1**
- **Lecture 6: Mining Data Streams 2**
- **Chapter 4** **(http://infolab.stanford.edu/~ullman/mmds/ch4.pdf)** in *Mining of Massive Datasets*, by Jure Leskovec, Anand Rajaraman, and Jeffrey D. Ullman, 3rd edition, Cambridge University Press, 2020 (**http://www.mmds.org/** **(http://www.mmds.org/)**)

Submission, Presentation and Demonstration

To submit your homework, you upload your solution in a zip file to Canvas. Canvas records the submission time. **Submission on time, i.e. before the deadline, is awarded with 3 exam bonus points if your homework is accepted.** Bonus will not be given, if you miss the deadline. **Your homework solution must include**

1. Source code if required (with comments);
2. Makefile or scripts to build and run;
3. Report (in PDF) with a short description of your solution, instructions how to build and to run, command-line parameters, if any (including default values), results, e.g., plots or screenshots.

Within a week after the homework deadline, **you present and demonstrate** your homework on your own laptop to a course instructor. A link under Syllabus will be provided to **book a time slot for presentation**.

Grading and Bonus Policy

The grade for a homework is **pass/fail**. If you submit report your homework on time and your solution is accepted, you will get **3 bonus points** on your first ID2222 exam whenever you take it. Some homework include **optional tasks for extra bonus**.